Faster R-CNN

Abstract

- Object Detection을 수행하는 SOTA 모델들은 모두 물체의 위치를 찾기 위해 Region Proposal Algorithm을 사용
- SPPnet이나 Fast R-CNN은 Region Proposal 계산을 Bottleneck으로 생각하여 이를 줄이는 방법으로 Running time을 줄임
- 이 논문에서 소개하는 RPN은 Fully Convolution Network이며, 물체의 경계와 확률 값을 픽셀 단위로 예측, 어텐션 메커니즘을 사용하여 주목해야 할 부분 명시
- Fast R-CNN의 탐지에서 사용되는 RPN은 고품질의 region proposal 생성을 위해 end-to-end 의 방식으로 훈련

1. Introduction

- 기존 객체 탐지 기술은 region proposal과 region-based CNN의 발전으로 성능이 향상됨
- Fast R-CNN은 region proposal을 제외하면 실시간 성능에 가까우나, Selective Search와 같은 기존 proposal 기법이 병목 현상을 유발
- 합성곱 공유를 통해 region proposal 비용 절감
- Faster R-CNN에서는 region proposal을 위한 새로운 RPN을 도입하여 Fast R-CNN과 합성곱 계층을 공유
- RPN은 fully convolutional network(FCN) 구조로, 객체 탐지 proposal을 end-to-end 학습 가능
- 앵커 박스(anchor boxes)를 활용하여 다양한 크기 및 종횡비의 객체를 효과적으로 예측

2. Related Work

Object Proposals

- Object proposal 기법은 super-pixels 그룹화, sliding windows 등을 기반으로 함

Deep Networks for Object Detection

- **R-CNN**: Region proposal을 기반으로 CNN을 훈련해 객체를 **분류**하지만, bounding box 예측은 수행하지 않음.
- **OverFeat**: Fully connected layer를 활용해 **객체 위치(localization)**를 예측하는 방식
- **MultiBox**: 여러 개의 불확실한 박스를 예측하는 네트워크로, region proposal을 생성해 R-CNN에서 활용.
- **DeepMask**: 객체 탐지가 아닌 segmentation proposal 학습을 목표로 개발됨

3. Faster R-CNN

- **Faster R-CNN**은 두 개의 모듈로 구성

1. 첫 번째 모듈: Region Proposal Network (RPN) → 객체가 있을 가능성이 높은 영역(region)을 제안하는 합성곱 신경망
2. 두 번째 모듈: Fast R-CNN 탐지기 → RPN이 제안한 영역을 사용하여 객체를 탐지

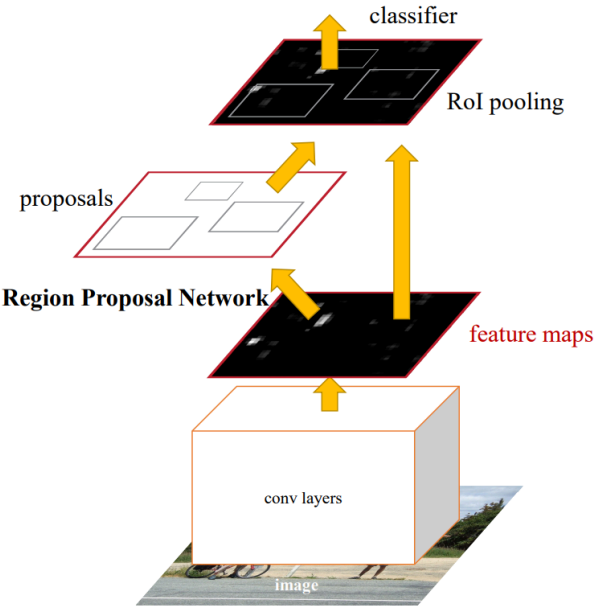

Figure 2: Faster R-CNN is a single, unified network for object detection. The RPN module serves as the 'attention' of this unified network.

- 단일 통합 네트워크로 동작, 중요한 부분 알려주는 attention 기능 사용

3.1 Region Proposal Networks

최종 목표는 **Fast R-CNN과 연산을 공유**하는 것이므로, 두 네트워크는 동일한 합성곱 계층을 공유한다고 가정

1. **합성곱 특성 맵 위에서 작은 네트워크(sliding network)를 적용**
    - 마지막 공유 합성곱 계층의 출력 특성 맵 위에서 작은 네트워크가 슬라이딩 방식으로 이동하며 region proposal을 생성
    - 이 네트워크는 n×n 크기의 영역을 입력으로 받아, 256차원(ZF) 또는 512차원(VGG) 특징 벡터로 변환
    - 활성화 함수로 **ReLU**를 사용.
2. **Fully Connected 계층(2가지 분기)**
    - 변환된 특징 벡터를 두 개의 sibling fully connected 계층에 입력:
        - **Box Regression Layer (reg)**: Bounding box 좌표를 예측
        - **Box Classification Layer (cls)**: 객체 존재 여부를 분류

- n=3 (3×3 합성곱 커널) 사용

3. **1×1 합성곱 연산으로 효율적인 구현**
    - Fully connected 계층을 슬라이딩 윈도우 방식으로 반복 적용하는 대신, n×n 합성곱 계층을 사용하여 연산량을 최적화.
    - 이후 1×1 합성곱 계층을 사용하여 bounding box 회귀(reg)와 분류(cls)를 위한 출력을 생성.

3.1.1 Anchors

- 각 슬라이딩 윈도우 위치에서 여러 개의 region proposal을 동시에 예측
- 한 위치에서 생성 가능한 최대 proposal 수를 k라고 하면,
    - Bounding Box Regression (reg) 계층 → 4k개의 출력 (각 bounding box 좌표 4개씩)
    - Classification (cls) 계층 → 2k개의 출력 (객체 여부 확률)
- k개의 proposal은 k 참조 박스를 기준으로 매개변수화 되며 이를 앵커(anchor)라고 부름

Translation-Invariant Anchors

- 이 방법의 중요한 특징은 변환 불변성을 가짐. 즉, 이미지에서 객체가 이동할 경우 proposal은 이동 되어야 하며 이는 같은 방법으로 각 위치에서 proposal을 예측
    - 이는 모델 크기가 작아 매개변수가 훨씬 적어 과적합 위험이 낮
- 기존 MultiBox 방식은 K-means를 이용해 800개의 앵커를 생성하지만, 이는 변환 불변성을 보장하지 못함

Multi-Scale Anchors as Regression References

- 기존 방식
    1. 이미지/특징 피라미드 사용 (연산량 증가)
    2. 여러 크기의 슬라이딩 윈도우 사용 (필터 크기 다양)
- 제안된 방식
    - 다양한 크기의 앵커 박스를 활용해 객체 분류, 경계 박스 회귀를 수행한다. 앵커 피라미드 방식을 적용하면 추가 연산 없이 피처를 공유할 수 있어 효율적

3.1.3 Training RPNs

- 256개의 Anchor를 임의로 뽑고, positive와 negative의 비율을 1:1로 맞춰서 Loss를 계산

3.2 Sharing Features for RPN and Fast R-CNN

- RPN(Region Proposal Network)과 Fast R-CNN을 각각 독립적으로 학습하면 서로 다른 방식으로 합성곱 계층이 조정(각각 다른 방향으로 학습)
- 따라서 두 네트워크가 합성곱 계층을 공유하도록 학습하는 방법이 필요
- 세 가지 학습 방식

1. **Alternating Training**
    - 먼저 RPN을 학습한 후, 여기서 추출한 Proposal 기반으로 Fast R-CNN을 학습
    - 이 과정에서 Fine-tuning이 일어나고, 그 Weight를 다시 RPN에 초기화
2. **Approximate Joint Training**
    - RPN과 Fast R-CNN을 하나의 네트워크로 합쳐 동시 학습
    - 공유된 Layer에서는 Loss가 두 방향에서 오고, RPN과 Fast R-CNN에서 Loss가 각각 오면서 합쳐짐
    - 역전파 시 두 네트워크의 손실을 함께 반영하지만, Proposal의 Box의 좌표값에 대한 미분을 무시하여 근사적인 방식
    - 성능 차이는 크지 않지만 학습 속도가 25~50% 향상됨.
3. **Non-approximate Joint Training**
    - Box 좌표도 gradients에 포함되도록 함
    - 박스 좌표에 대해 미분 가능한 RoI 풀링 레이어가 필요, "RoI warping" 층에 의해 해결책이 제시

4단계 교대 학습 (4-Step Alternating Training)

1. RPN을 먼저 학습 (ImageNet 사전 학습 모델로 초기화 후 미세 조정)
2. Fast R-CNN을 Detector로 하여 RPN에서 뽑아낸 Proposal을 통해 학습
3. Convolution Layer는 고정시키고, RPN에 대해 고유한 Layer만을 학습
4. Convolution Layer를 유지한 채, Fast R-CNN 부분을 학습

    → 추가 반복을 해도 성능 향상은 미미함

3.3 Implementation Details

- 학습과 테스트 모두 단일의 크기를 가진 영상을 입력으로 사용
- 사진 한 장에 약 20000개 정도의 Anchor가 사용되는데, 경계를 넘는 것을 제외하면 6000개 정도가 남
- 테스트 시,전체 이미지에 RPN을 적용하며, 경계를 넘는 제안 박스는 이미지 내부로 클리핑
- RPN을 통해 뽑아낸 Proposal은 겹치는 부분이 많이 존재함, 이를 줄이기 위해 Class Score에 기반한 Non-maximum Suppression (NMS)를 도입